Average Path Length as a Paradigm for the Fast Evaluation of Functions Represented by Binary Decision Diagrams*

T. Sasao†, J. T. Butler‡, and M. Matsuura†

†Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka, Fukuoka, 820-8502 JAPAN, sasao@cse.kyutech.ac.jp and matsuura@cse.kyutech.ac.jp

‡Dept. of Electrical and Computer Eng., Naval Postgraduate School, Code EC/Bu, Monterey, CA 93943-5121, USA, jbutler@nps.navy.mil

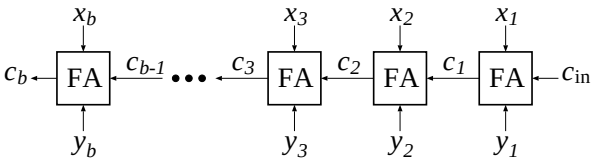
November 2, 2002

Abstract— This paper focuses on the average path length (APL) of BDD's for switching functions. APL is a metric for the time it takes to evaluate the function by a computer program. We derive the APL for the AND, OR, parity, carry-out, comparison, threshold symmetric, and majority functions. We also consider the average of the APL for various classes of functions, including symmetric, threshold symmetric, and unate cascade. For symmetric functions, we show the average APL is close to the maximum path length, n, the number of variables. We show there are exactly two functions, the parity functions, that achieve the upper bound, n, on the APL for BDD's over all functions dependent on n variables. All other functions have an APL strictly less than n. We show that the APL of BDD's over all functions dependent on n variables is bounded below by $2 - \frac{1}{2^{n-1}}$. The set of functions that achieves this small value is uniquely the set of unate cascade realizable functions. We also show that the APL for benchmark functions is typically much less than for random functions.

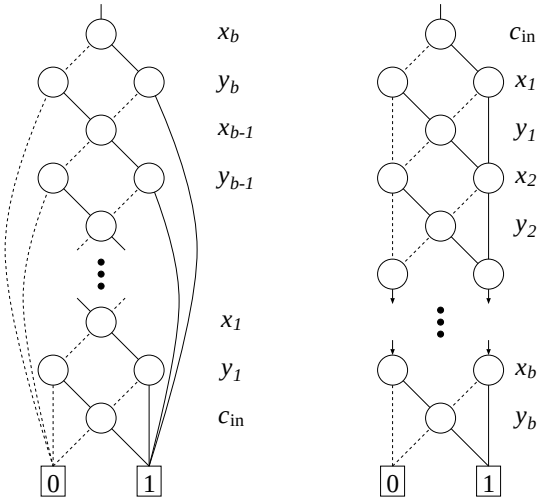
I. INTRODUCTION

Within the past 25 years, considerable research has been devoted to minimizing the number of nodes in binary decision diagrams (BDD's). This problem is motivated by the fact that the amount of memory needed to store a function as a BDD is directly proportional to the number of nodes. However, there is another cost, the cost of evaluating the function for some combination of variable values. This is the subject of this paper.

We assume that, for a given function f, all 2^n assignments of values to the n variables are equally likely. Thus, the average path length (APL) in the BDD of f can be computed by summing the path lengths over all 2^n assignments of values to the n variables and dividing by 2^n. We are also interested in the distribution of path lengths, namely the number paths for each path length value. Therefore, we can determine the extent to which pathological cases occur. To illustrate, consider the BDD of the carry-out function of a b-bit ripple carry adder, where $n = 2b + 1$. That is, we assume each bit to be added contributes two input variables, and there is one carry-in variable. Fig. 1a shows the ripple-carry adder circuit that produces this function. Fig. 1b shows the BDD in which the variables are in descending order (most significant bits at the top), and Fig. 1c shows the BDD in which the variables are in ascending order (least significant bits at the top). Note that a dotted edge corresponds to $x_i = 0$ and a solid line corresponds to $x_i = 1$. Although both orderings yield BDD's with the same number of nodes, the first has more shorter paths, as shown by the distributions in Fig. 2 for $n = 33$. The sawtooth pattern associated with the most significant bit (MSB) at the top occurs because there are no paths of odd length. In this distribution, most paths have lengths less than 10. The distribution associated with the least significant bit



(a) Carry-out c_b of b-bit ripple carry adder circuit.



(b) BDD for c_b with MSB at top. (c) BDD for c_b with LSB at top.

Fig. 1. Two BDD's for the carry-out function of a b-bit ripple carry adder.

*Research supported by the Takeda Research Foundation.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
02 NOV 2002		

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Average Path Length as a Paradigm for the Fast Evaluation of Functions Represented by Binary Decision Diagrams	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Postgraduate School,Department of Electrical and Computer Engineering,Monterey,CA,93943**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper focuses on the average path length (APL) of BDD?s for switching functions. APL is a metric for the time it takes to evaluate the function by a computer program. We derive the APL for the AND, OR, parity, carry-out, comparison threshold symmetric, and majority functions. We also consider the average of the APL for various classes of functions including symmetric, threshold symmetric, and unate cascade. For symmetric functions, we show the average APL is close to the maximum path length, n, the number of variables. We show there are exactly two functions, the parity functions, that achieve the upper bound, n, on the APL for BDD?s over all functions dependent on n variables. All other functions have an APL strictly less than n. We show that the APL of BDD?s over all functions dependent on n variables is bounded below by 2 ? 1 2n?1 . The set of functions that achieves this small value is uniquely the set of unate cascade realizable functions. We also show that the APL for benchmark functions is typically much less than for random functions.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

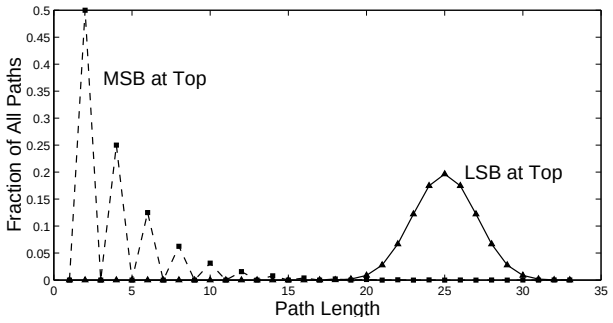


Fig. 2. Distributions of path lengths for the carry-out function.

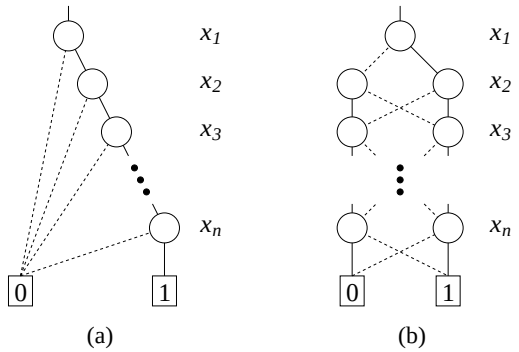


Fig. 3. BDD's for the AND and parity functions.

(LSB) at the top is binomial and centered around path length 25. Thus, the APL for the LSB at the top is 25, which is significantly more than the APL for the MSB at the top, 4.0. In pass transistor implementations [1] [2] of BDDs, significant differences in APL result in significant differences in delay in logic circuits.

The path length in decision diagrams is important in databases [3], pattern recognition [4], and analysis of algorithms; for example, decision trees have been used to determine lower bounds on the complexity of sorting algorithms [5]. [6] is an easily understood (but old) survey. Recently, Liu, Wang, Hwang, and Liu [7] show three heuristics for finding the minimal APL in BDD's. They show that the ordering of variables that achieves 1. the minimum path length and 2. the minimum size (number of nodes) is different for all of the 20 benchmark functions they considered.

Our results extend the results of Brandman, Orlitsky, and Hennessey [8] in which lower bounds on the APL are computed. In our paper, we compute exact values for the APL, as well as distributions of path lengths. We show that some distributions, like that of the AND function, extend over the whole range of possible path lengths, while others, like that of the parity functions, are very narrow.

Because of space limitations, we do not include proofs. For a complete version of this paper, see [9]

II. Individual Functions

In this section, we consider the path lengths in BDD's associated with individual functions. Our approach is to use generating functions to track the number of paths of various lengths.

A. AND and OR Function

Fig. 3a shows the BDD of an n-variable AND function. The generating function for the distribution of path lengths in the (trivial) BDD of the AND function on one variable is $2z^1$. That is, there are two assignments of variables ($x_1 = 0$ and $x_1 = 1$) each corresponding to a path of length 1. The AND function of two variables has a path length distribution described by $2z^1 + 2z^2$. In this case, two assignments of values cause the AND to be evaluated as 0 through a path of length 1 ($x_1x_2 = 00$ and 01) and two assignments ($x_1x_2 = 10$ and 11) cause the function to be evaluated as 0 or 1 through a path of length 2. For the n-variable AND, the path length distribution, $PLD(AND(n))$ is

$$\begin{aligned} PLD(AND(n)) &= 2^{n-1}z + 2^{n-2}z^2 + \cdots + 2^2 z^{n-2} + \\ &\quad 2^1 z^{n-1} + 2 \cdot 2^0 z^n \\ &= \frac{2^n - \frac{z^{n+1}}{2}}{1 - \frac{z}{2}} + z^n - 2^n. \end{aligned}$$

Dividing this by 2^n yields, $FPLD(AND(n))$, a generating function for the fraction of paths of length 1, 2, etc..

$$FPLD(AND(n)) = \frac{1 - \frac{z^{n+1}}{2^{n+1}}}{1 - \frac{z}{2}} + \frac{z^n}{2^n} - 1. \qquad (1)$$

The APL for the AND function, $APL(AND(n))$ is calculated by forming the weighted sum of path lengths and dividing by 2^n, the number of assignments of values to the n-variables. This can be done by differentiating $PLD(AND(n))$ with respect to z, multiplying by z, and setting $z = 1$. For example, from the discussion above, $PLD(AND(2)) = 2z^1 + 2z^2$. Differentiating with respect to z and multiplying by z yields $2z^1 + 4z^2$ which has the effect of generating the weighted sum associated with path of lengths 1 ($2z^1$) and 2 ($4z^2$). Setting $z = 1$ forms the sum over all weighted values, which is 6 in this case. Dividing by 2^2, the number of assignments of values to two variables, yields $APL(AND(2)) = 1.5$. Performing these steps on (1) yields a result by Breithart and Gal [10]

Lemma 2.1.

$$APL(AND(n)) = APL(OR(n)) = 2 - \frac{1}{2^{n-1}}.$$

The above lemma also applies to the OR function, because its BDD is isomorphic to that of the AND function.

B. Parity Function

Fig. 3b shows the BDD for a parity function (the exclusive OR). Here, all paths from the root node to a terminal node include a node associated with each variable. Since there are no "short-cuts", the length of all paths is n, and the distribution of nodes is $2^n z^n$.

Lemma 2.2.

$$APL(PARITY(n)) = n.$$

C. Functions With the Largest and Smallest APL

In a function whose APL is exactly n, all paths must have maximum length. Functions with this property are rare, as shown below.

Lemma 2.3. *The parity functions are uniquely those functions whose BDD's have the largest APL (n) among all n-variable functions.*

The AND and the OR functions also have a special distinction.

Definition 2.1. *f is a unate cascade realizable function if f can be represented as*

$$f(x_1, x_2, \ldots x_n) = x_1^* \diamondsuit_1 (x_2^* \diamondsuit_2 (\ldots (x_{n-1}^* \diamondsuit_{n-1} x_n^*)) \ldots),$$

where x_i^* is either x_i or $\bar{x}_i$ and $\diamondsuit_i$ is either the OR $\vee$ or AND $\wedge$ function.

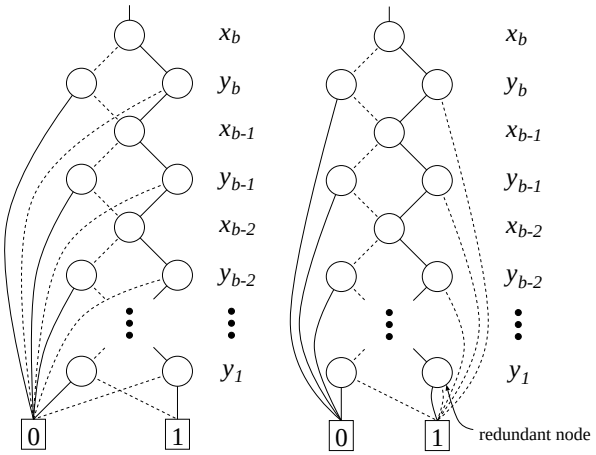
For example, if all variables occur uncomplemented and all operations $\diamondsuit_i$ are AND (respectively, OR), the resulting function is the AND (respectively, OR) of all variables. The significance of a unate cascade realizable function is that its BDD has the property that all nodes have at least one edge from that node to a terminal node. Further, for any such BDD, we can find a unate cascade realizable function corresponding to it.

Lemma 2.4. *The unate cascade realizable functions are uniquely those functions whose BDD's have the smallest ($2 - \frac{1}{2^{n-1}}$) APL among all n-variable functions.*

D. Carry-out and Comparison Functions

In this section, we consider the carry-out, $X = Y$ (equal to), and $X \geq Y$ (greater than or equal to) functions. Fig. 4 shows the BDD's for the $X \geq Y$ and $X = Y$ function, where the MSB is at the top. Fig. 1 shows the BDD for the carry-out function.

In the case of the $X = Y$ function, the position of the pair $(x_i y_i)$ in the ordering is irrelevant. However, for the carry-out and $X \geq Y$ functions, the smallest APL occurs when the MSB is at the top. For example, in the $X \geq Y$ function, the MSBs are *always* needed to determine whether the output is 1 or 0. Suppose that, the lowest MSB from the top variable



(a) $X = Y$ function (b) $X \geq Y$ function

Fig. 4. BDD's for the $X = Y$ and $X \geq Y$ functions.

(either x_b or y_b) is the k-th variable from the top, and assume that $k > 2$. The two MSBs are needed to determine the function value. Further, the other ($k - 2 > 0$) variables are needed if $x_b = y_b$. It follows that all paths from the root node to a constant node have length k. We can state.

Lemma 2.5. *A BDD for the carry-out and $X \geq Y$ functions of smallest APL is achieved by placing the MSBs at the top, the next most significant bits next, etc..*

As in the previous examples, we derive a generating function for the distribution of paths to path lengths, and, from this, derive the APL. The results are shown in Table I. It is interesting that all three functions have an APL that is a constant 4 for large n. Except for the bottom, the BDD's structure is the same for all functions. Indeed, the difference in the values is explained entirely by the structure at the bottom.

TABLE I
DISTRIBUTION OF PATH LENGTH AND APL FOR COMPARISON FUNCTIONS

Function	Distribution	APL
Carry-out	$2^{2b} z^2 (1 - \frac{z^{2b}}{2^b})/(1 - \frac{z^2}{2}) + 2^{b+1} z^{2b+1}$	$4 - \frac{3}{2^b}$
$X = Y$	$(2^{2b-1} z^2 - 2^b z^{2b})/(1 - \frac{z^2}{2}) + 2^{b+1} z^{2b}$	$4 - \frac{4}{2^b}$
$X \geq Y$	$(2^{2b-1} z^2 - 2^{b-1} z^{2b})/(1 - \frac{z^2}{2}) + 2^b z^{2b-1}$	$4 - \frac{5}{2^b}$

E. Symmetric Threshold Function

A *symmetric function* has the property that permuting any of the variables leaves the function unchanged. A *symmetric threshold function* is a symmetric function that is 1 iff t of the input variables are 1, for $0 \leq t \leq n$. The *majority function*

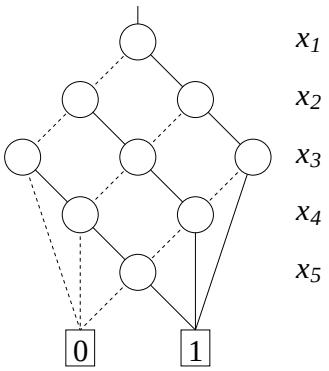


Fig. 5. BDD of the 5-variable majority function

is a special case of a symmetric threshold function, where $n = 2t - 1$ and $t = 1,\ 2,\ \ldots$.

For this class of functions, we can state

Lemma 2.6. *The generating function for the distribution of the fraction of paths with various path lengths for the symmetric threshold function with threshold t is*

$$FPLD(S\_THRES(n,t)) = \sum_{j=k}^{n} \left[\binom{j-1}{k-1} + \binom{j-1}{n-k} \right] \left(\frac{z}{2}\right)^j,$$

where $k = \min\{t, n-t+1\}$.

Lemma 2.7.

$$APL(S\_THRES(n,t)) = 2k - \sum_{j=1}^{k} \frac{\binom{n-j}{k-j}}{2^{n-j}} j.$$

where $k = \min\{t, n-t+1\}$,

and when n is large, we can write

Lemma 2.8.

$$APL(S\_THRES(n,t)) \sim 2\min\{t, (n-t+1)\},$$

where $A(n) \sim B(n)$ means $\lim_{n\to\infty} \frac{A(n)}{B(n)} = 1$.

F. Majority Function

A majority function is a special case of a symmetric threshold function. Specifically, $n = 2t - 1$, where t is the threshold. Fig. 5 shows the BDD of the majority function on 5 variables. In the case of a symmetric threshold function, we assumed that the threshold t stayed fixed as n, the number of variables, increased. In the case of the majority function, we assume that t increases as n increases (so that the relation $n = 2t - 1$ always holds). We have

Lemma 2.9.

$$FPLD(MAJ(n)) = 2^t z^t \sum_{i=0}^{t-1} \binom{t-1+i}{t-1} \left(\frac{z}{2}\right)^i. \quad (2)$$

where $n = 2t - 1$, for $t = 1, 2, \ldots$.

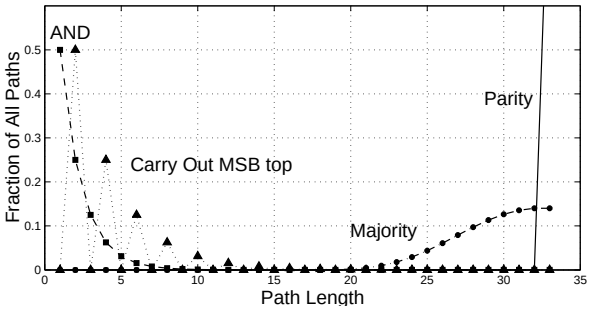


Fig. 6. Distributions of path lengths for various functions

Lemma 2.10.

$$APL(MAJ(n)) = n - \frac{n+1}{2^n}\binom{n}{\frac{n-1}{2}} + 1, \quad (3)$$

where n is odd.

The second term in (3) contains the factor $\binom{n}{\frac{n-1}{2}}$, which is $\frac{n!}{\frac{n-1}{2}!\frac{n+1}{2}!}$. The factorials can be approximated using Sterling's approximation, yielding

Lemma 2.11.

$$APL(MAJ(n)) \sim n - \sqrt{\frac{2n}{\pi}} + 1.$$

Here, we retain the term +1 because it is significant for small values of n. Indeed, this approximation is quite accurate for small values of n. For example, for $n = 3$, the exact expression and the approximation differ by less than 5%, while for $n = 15$, the difference is only 0.4%.

G. Comparison of Distributions

Fig. 6 shows the distributions of path lengths for the AND, majority, and parity functions on $n = 33$ variables. The path lengths in the BDD of the AND function range from 1 to 33, while the path length distribution of the parity function is a single point, at $n = 33$. In between, is the majority path length distribution.

III. APL FOR SETS OF FUNCTIONS

In this section, we consider the APL over sets of functions. We assume that each function contributes equally to this "average of averages". We consider four sets of functions, all functions, all (totally) symmetric functions, all symmetric threshold functions, and all unate cascade realizable functions.

Definition 3.1. *$APL_S(n)$ denotes the average path length of BDD's for n-variable functions in the set S. That is, $APL_S(n) = \frac{1}{|S|}\sum_{f_i \in S} APL(f_i)$, where S denotes a set of functions.*

TABLE II
$APL_{all}(n)$ FOR $3 \leq n \leq 16$ OBTAINED BY AVERAGING 10,000 SAMPLES.

n	$APL_{All}(n, \Pi)$
3	†2.187500
4	†3.183594
5	4.226113
6	5.203656
7	6.195206
8	7.188487
9	8.186332
10	9.184930
11	10.184335
12	11.183719
13	12.183499
14	13.183635
15	14.183664
16	15.183647

†Exact value obtained by enumerating all functions.

A. Set of All Functions

In this section, we determine an upper bound on the APL for all functions on n variables. Let $APL_{All}(n, \Pi)$ be the APL over all n-variable functions for *fixed ordering* Π of the variables.

Lemma 3.1.

$$APL_{All}(n, \Pi) = APL_{All}(n-1, \Pi) + 1 - \frac{1}{2^{2^{n-1}}}.$$

Theorem 3.1.

$$APL_{All}(n) \leq (n-1) + 0.183578.$$

Table II shows the APL for randomly generated functions on n-variables, for $3 \leq n \leq 17$. For $n = 3$ and $n = 4$, exact values are obtained by enumerating all functions. For each value of n in the range $5 \leq n \leq 14$, 10,000 samples were generated.

Note that this data was generated by choosing some fixed ordering of the variables, and then computing the APL through the sample function. This data matches closely the upper bound.

B. Set of All Functions With a Specified Number of Minterms

We now consider how the number of true minterms in a function affects its APL. That is, instead of considering all n-variable functions as a single set, we divide this set into subsets, according to the number true minterms. Then, we seek the APL of all functions in each subset. Table III shows $APL_{all}(n, k, \Pi)$ the average APL of BDD's for n-variable functions with k true minterms. Each data value is the average of 10,000 random samples. The APL values are symmetric about the middle k value, since the BDD of a function with k true minterms is isomorphic to the BDD of its complement, which has $n - k$ true minterms. We have omitted this mirror-image data.

TABLE III
$APL_{all}(n, k, \Pi)$ FOR $n = 6,\ 8,$ and 10 VARIABLES.

$k/2^n$	$APL_{all}(n, k, \Pi)$		
	$n = 6$	$n = 8$	$n = 10$
0.0000	0.000000	0.000000	0.000000
0.0625	3.364556	5.246588	7.218377
0.1250	4.152775	6.097209	8.082283
0.1875	4.578462	6.540764	8.533054
0.2500	4.844106	6.816895	8.810673
0.3125	5.013888	6.995677	8.989050
0.3750	5.121319	7.107553	9.101985
0.4375	5.184331	7.169411	9.164830
0.5000	5.203656	7.188487	9.184930

C. Set of All Symmetric Functions

Moret, Thomason, and Gonzalez [11] show that a BDD of any symmetric functions dependent on n variables has at least one path that is the longest possible, n. In this section, we extend this by deriving the average APL of BDD's of symmetric functions.

Lemma 3.2.

$$APL_{Sym}(n) = APL_{Sym}(n-1) + 1 - \frac{1}{2^n}$$

Theorem 3.2.

$$APL_{Sym}(n) = n - 1 + \frac{1}{2^n}.$$

D. Set of All Symmetric Threshold Functions

A symmetric threshold function is a symmetric function that is 1 iff k or more of the variables are 1. For example, the OR and AND functions on n variables are symmetric threshold functions with $k = 1$ and $k = n$, respectively. We have

Theorem 3.3.

$$APL_{SymThres}(n) = \frac{n^2 + n}{2(n+2)}.$$

IV. BENCHMARK FUNCTIONS

The APL of a randomly generated function of n variables is near n. It is interesting to consider the APL for benchmark

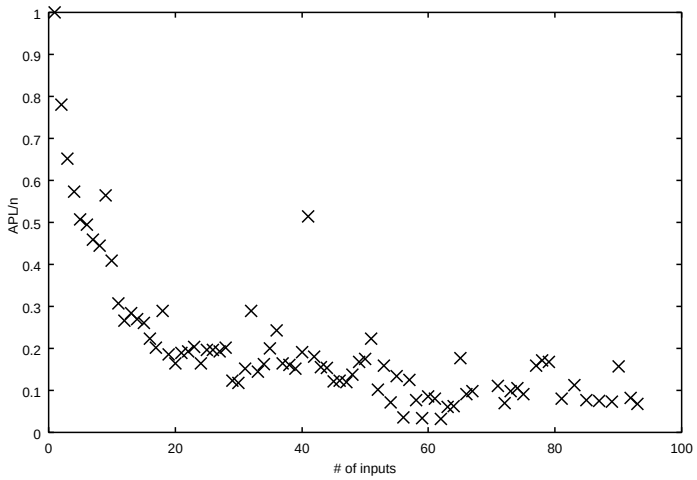


Fig. 7. Graph of $\frac{APL}{n}$ versus n for 4352 benchmark functions.

functions. To investigate this, we considered 189 benchmark functions in [12]. For each output f_j of a multiple-output function, we minimized the number of nodes in the BDD for f_j (using an exhaustive method for $n \leq 16$ and a heuristic method for $16 < n$) and computed the resulting APL. In all, we minimized 4352 single-output functions. For each n, we collected the functions that depend on n variables, and obtained the average of their APL. To compare among various values of n, we divided this average by n.

The result is shown in the graph of Fig. 7, in which the value of $\frac{APL}{n}$ is plotted verses n. Each $\times$ in this figure corresponds to the average of all n-variable benchmark functions normalized by dividing by n. Note that $\frac{APL}{n}$ is much smaller than 1.0 except for small values of n and $n = 41$. The outlier at $n = 41$ is due to function C499, used for single error correction, which has 32 outputs on 41 variables, each with an APL of approximately 25. There is a tendency for $\frac{APL}{n}$ to decrease as n increases. This is to be compared to randomly generated functions, where $\frac{APL}{n}$ tends to increase as n increases. This shows that the benchmark functions have quite different properties than randomly generated ones. Indeed, since we minimized the node count and not the APL, we would expect an even greater difference had we minimized the APL.

V. Concluding Remarks

In this paper, we consider the average path length or APL of BDD's, as measured by the average number of edges traversed from the root node to a terminal node. We have shown that the APL, for some functions, such as the carry-out, is strongly dependent on the ordering of the variables. We have derived the APL for various classes of functions. This is summarized in Table IV

TABLE IV
Average path length in BDD's of sets of functions for large n.

Function	APL
PARITY	n*
AND and OR	$2 - \frac{1}{2^{n-1}}$*
Symmetric Threshold k	$2k$
MAJ (Majority)	$n - \sqrt{\frac{2n}{\pi}}$
BTREE (Balanced Tree)	$\log_2 n$
Carry-out	$4 - \frac{5}{4^n}$
All	$\leq n - 0.816$
All Symmetric	$n - 1 + \frac{1}{2^n}$
All Symmetric Threshold	$\frac{n^2+n}{2(n+2)}$*
All Unate Cascade	$2 - \frac{1}{2^{n-1}}$*

* Exact value.

References

[1] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design," *IEEE J. Solid State Circuits*, vol. 31, pp. 792–803, June 1996.

[2] M. Tachibana, "Heuristic algorithms for FBDD node minimization with application to pass-transistor-logic and dcvs synthesis," *SASHIMI'96*, pp. 96–101, November 1996.

[3] M. Hanani, "An optimal evaluation of Boolean expressions in an online query system," *Communications of the ACM*, vol. 20, pp. 344–347, May 1977.

[4] D. A. Bell, "Decision trees, tables, and lattices," *Pattern Recognition: Ideas in Practice*, vol. Chapt. 5, B. G. Batchelor, Ed., Plennnum Press, New York 1978.

[5] D. E. Knuth, "Mathematical analysis of algorithms," *Proc. IFIP Congress 71*, vol. 1, pp. 135–143, Addison-Wesley, Reading, Mass. 1971.

[6] B. N. E. Moret, "Decision trees and diagrams," *Computing Surveys*, vol. 14, pp. 593–623, December 1982.

[7] Y. Y. Liu, K. H. Wang, T. T. Hwang, and C. Liu, "Binary decision diagram with minimum expected path length," *DATE2001*, pp. 1–5, March 2001.

[8] Y. Brandman, A. Orlitsky, and J. Hennessey, "A spectral lower bound for the size of Boolean trees and two-level AND-OR circuits," *IEEE Trans. on Comput.*, vol. 39, pp. 282–287, Februrary 1990.

[9] T. Sasao, J. T. Butler, and M. Matsuura, "Average path length as a paradigm for the fast evaluation of functions represented by binary decision diagrams," *NPS Technical Report*, 2002.

[10] Y. Breithart and S. Gal, "Analysis of algorithms of the evaluation of monotonic Boolean functions," *IEEE Trans. on Comput.*, vol. C-27, pp. 1083–1087, November 1978.

[11] B. N. E. Moret, M. G. Thomason, and R. C. Gonzalez, "Symmetric and threshold Boolean functions are exhaustive," *IEEE Trans. on Comput.*, vol. C-32, pp. 1211–1212, December 1983.

[12] "http://www.cbl.ncsu.edu/benchmarks/," *The Benchmark Archives at CBL*.